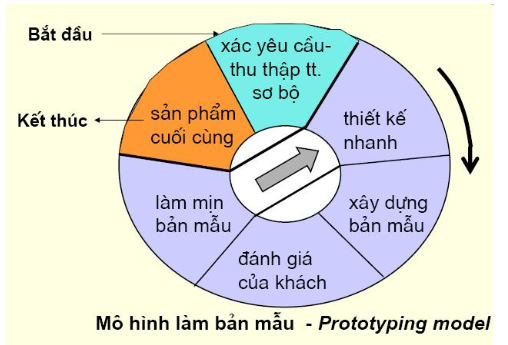
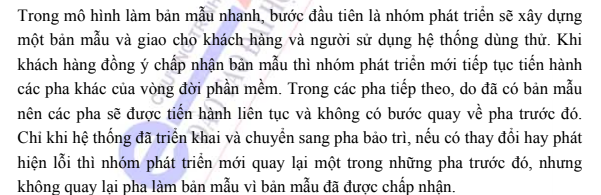
**Phân tích thiết kế hệ thống thông tin**

**Đề 1: AT11E**

**Câu 1: (2đ) Anh/chị hãy trình bày quá trình phát triển phần mềm theo mô hình bản mẫu? Nêu rõ ưu, nhược điểm và phạm vi áp dụng của mô hình này?**





**Ưu điểm:**

Nhanh chóng xác định được yêu cầu

Tạo cơ sở ký hợp đồng

Giúp đào tạo huấn luyện người sử dụng

**Nhược điểm:**

Tính cấu trúc không cao

Khách hàng ít tin tưởng

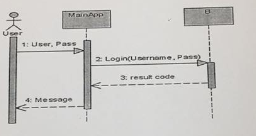
**Phạm vi áp dụng:**

Các yêu cầu chưa rõ rang

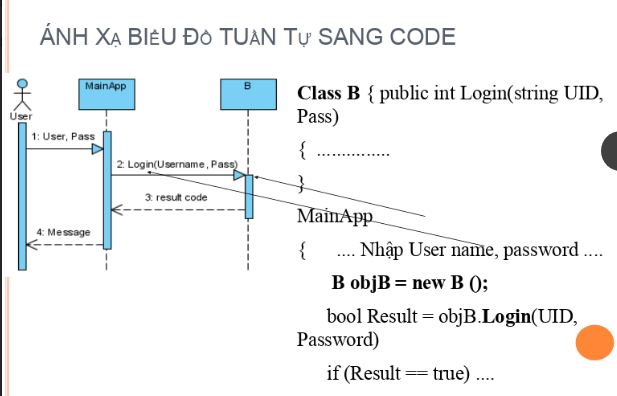
Input/output chưa rõ rang

Khó đánh giá hiệu quả của thuật toán

**Câu 2: (3đ) Cho biểu đồ tuần tự như hình vẽ:**

****

**Anh/chị hãy phân tích ý nghĩa của biểu đồ trên và minh họa sự tương tác giữa các class trong mô hình này bằng một ngôn ngữ lập trình hướng đối tượng?**

 **(JAVA)**

Giải:

Biều đồ tuần tự như hình vẽ :

* Ý nghĩa của biểu đồ trên:
* User: người sử dụng sẽ nhập username, password ở class MainApp.
* Sau đó, class MainApp sẽ gửi thực hiện login(username,password) ở class B.
* Class B sẽ trả kết quả code về cho class MainApp
* Class MainApp sẽ gửi thông tin đăng nhập cuối cho user.
* Tương tác giữa các class:
  + - User sẽ được sử dụng class MainApp để đăng nhập thông tin.
    - Class MainApp đóng vai rò như controlle(trung gian) cho việc giao tiếp giữa class B và User.
    - Class B sẽ trả thông tin kết quả về cho class MainApp
* Code

Class B{

Public Boolean login(String Username, String Password)

{

//xử lý đăng nhập

If(tài khoản đúng){ //TH đăng nhập đúng

Return true;}

Else(tai khoản sai){ //TH đăng nhập sai

Return false;}

}

}

Class MainApp{

Public static void main (String[] agrs)

{

B obj B=new B(); // khởi tạo obj B

String Username; //Nhập username

String Password; //Nhập password

Boolean result = B.login(username,password)

If(result==true)

{ //in ra message đăng nhập thành công}

Else

{//in ra message đăng nhập thất bại}



**Câu 3: (5đ) Cho đặc tả yêu cầu hệ thống quản lý thư viện như sau:**

* **Hệ thống cho phép người sử dụng biết sách hiện có, và có thể đặt mươn 2 quyển sách. Những người tham gia mượn sách có account để truy cập hệ thống và có thể biết sác nào đã mượn hoặc đã đặt.**
* **Bên cạnh đó, việc mượn sách được thực hiện bởi các thủ thư, sau khi xác định người mượn sách, họ biết được người này có được phép mượn hay không? (tối đa 5 quyển) , người này có được ưu tiên hay không? (đã đặt trước)”**

1. **Anh/chị hãy xác định các actors và chức năng của hệ thống trên. Vẽ biểu đồ usecase tương ứng của hệ thống.**
2. **Anh/chị hãy mô tả chi tiết một usecase nào đó trong biểu đồ trên.**

Giải:

1. Hệ thống có 2 actors:

+ Người mượn sách

+ Người thủ thư

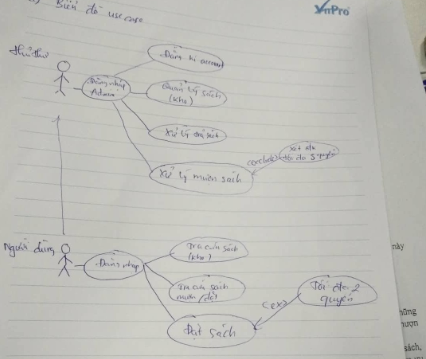
Chức năng hệ thống:

+ Đăng nhập

+ Xem kho sách

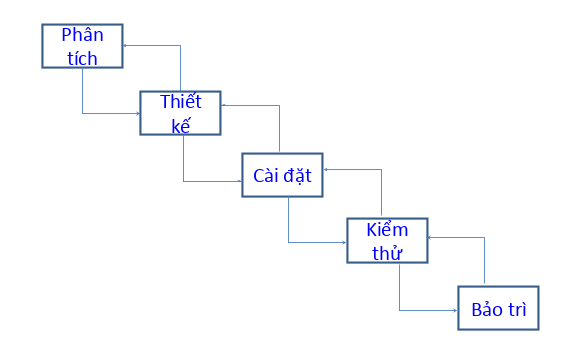
+ Đặt mượn sách

+ Xem sách đã mượn



**Đề 2:**

**Câu 1: Anh/chị hãy trình bày quá trình phát triển phần mềm theo mô hình thác nước? Nêu rõ ưu, nhược điểm và phạm vi áp dụng mô hình này?**



**Quá trình:**

* **Thu thập và phân tích thông tin về yêu cầu phần mềm, chức năng,**

**hiệu năng, giao diện.**

* **Tài liệu đặc tả để khách hàng duyệt lại và làm tài liệu cho người**

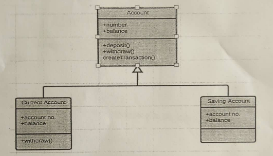
**phát triển.**

**Thiết kế: là quá trình chuyển hóa yêu cầu phần**

**mềm thành các mô tả thiết kế**

* Ưu điểm:
* Có vòng lặp, cho phép trở về pha trước để sửa chữa khi phát hiện lỗi hoặc khi có thay đổi.
  + - Dễ bảo trì: tất cả các pha trong chu trình đều được viết tài liệu cẩn thận và được kiểm tra trước khi chuyển sang pha tiếp theo.
* Khuyết điểm:
* Sản phẩm phần mềm cuối cùng có thể không thỏa mãn nhu cầu thực sự của khách hàng.
* Mô hình Thác nước SDLC được sử dụng khi:
* Yêu cầu ổn định và không thay đổi thường xuyên.
* Một ứng dụng nhỏ.
* Không có yêu cầu mà không hiểu hoặc không rõ ràng.
* Môi trường ổn định
* Các công cụ và công nghệ được sử dụng là ổn định
* Nguồn lực được đào tạo và sẵn sàng.

**Câu 2: (3đ) Cho biểu đồ lớp như hình vẽ:**



**Anh/chị hãy phân tích ý nghĩa của biểu đồ trên và minh họa sự tương tác giữa các class trong mô hình này bằng một ngôn ngữ lập trình hướng đối tượng? (JAVA)**

**Giải:**

* **Phân tích:**
* UC gốc ( Account) là một UC hoàn thiện;
* Cả hai UC (CurrentAccount và SavingAccount) sử dụng lại balance của Uc tổng, UC CurrnetAccount sử dụng lại withdraw();
* Trong cùng một điều kiện, một UC được mở rộng bằng UC khác.
* **Code:**

Class Account{

Public int number;

Public int balance;

Public void deposit();

Public void withdraw();

Public void createTransaction();

}

Class CurrentAccount extends Account{

Public int accountNo;

Public int balance;

Public void withdraw();

}

Class SavingAccount extends Account{

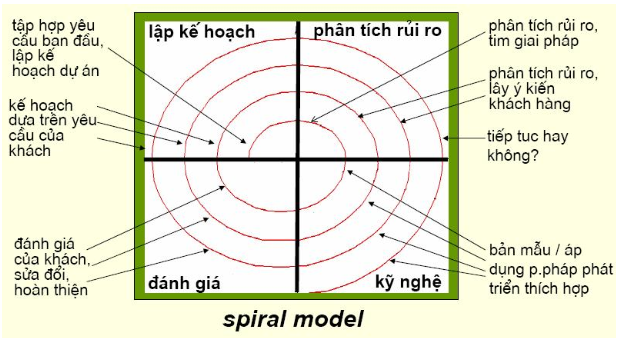
Public int accountNo;

Public int balance;

}

**Đề 3:**

**Câu 1: (2đ) Anh/chị hãy trình bày quá trình phát triển phần mềm theo mô hình xoắn ốc? Nêu rõ ưu, nhược điểm và phạm vi áp dụng của mô hình này?**



**Qúa trình phát triển phần mềm**

Mô hình xoắn ốc có thể được xem là sự kết hợp giữa mô hình thác nước và mô hình mẫu và đồng thời thêm một thành phần mới - phân tích rủi ro.  
  
Trong mô hình xoắn ốc, quy trình phát triển phần mềm được biểu diễn như một vòng xoắn ốc. Các pha trong quy trình phát triển xoắn ốc bao gồm:

* Thiết lập mục tiêu: xác định mục tiêu cho từng pha của dự án.
* Đánh giá và giảm thiểu rủi ro: rủi ro được đánh giá và thực hiện các hành động để giảm thiểu rủi ro.
* Phát triển và đánh giá: sau khi đánh giá rủi ro, một mô hình xây dựng hệ thống sẽ được lựa chọn từ những mô hình chung.
* Lập kế hoạch: đánh giá dự án và pha tiếp theo của mô hình xoắn ốc sẽ được lập kế hoạch.

Ưu điểm

* Phân tích rủi ro dự án được đầy lên làm một phần thiết yếu trong quy trình xoắn ốc để tăng độ tin cậy của dự án
* Xây dựng dự án có sự kết hợp các mô hình khác vào phát triển (Thác nứơc, mô hình mẫu…)
* Cho phép thay đổi tuỳ theo yêu cầu cho mỗi vòng xoắn ốc
* Nó được xem như là một mô hình tổng hợp của các mô hình khác. Không chỉ áp dụng cho phần mềm mà còn phải cho cả phần cứng
* Một rủi ro nào đó không được giải quyết thì chấm dứt dự án
* Các vòng tròn được lặp để đáp ưng được những thay đổi của người dùng
* Kiểm soát rủi ro ở từng giai đoạn phát triển.
* Đánh giá tri phí chính xác hơn các phương pháp khác

Nhược điểm:

* Phức tạp và không thích hợp với các dự án nhỏ và ít rủi ro.
* Cần có kỹ năng tốt về phân tích rủi ro.
* Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn
* Chưa được dùng rộng dãi như mô hình thác nước hay là mẫu.
* Đòi hỏi năng lực quản lý

**Phạm vi:** Hợp với các hệ thống lớn có thể phân chia thành nhiều thành phần. Chưa được áp dụng rộng rai như mô hình thác nước và mẫu thử.

**Câu 3: Cho đặc tả yêu cầu hệ thống quản lý thư viện như sau:**

**Hiện nay các bạn sinh viên An toàn thông tin không những tham gia tích cực tron việc học tập trên lớp mà cò tích cực trong công tác xã hội (tình nguyên, văn thể,…) để quản lý thời gian của sinh viên, hệ quản lý muốn xây dựng một phần mềm quản lý thời gian mỗi sinh viên. Hệ thống cho phép:**

* **Cán bộ quản lý hệ theo dõi thời gian sinh hoạt xã hội, thời gian xin nghĩ của sinh viên(do ốm, bận,việc khác) và phê duyệt đơn xin nghỉ.**
* **Mỗi sinh viên có thể tự xem thời gian biểu của mình, nộp đơn xin nghỉ, và nhận email thông báo của cán bộ quản lý.**

1. **Anh/chị hãy xác định các actors và chức năng của hệ thống trên. Vẽ biểu đồ usecase tương ứng của hệ thống.**
2. **Anh/chị hãy mô tả chi tiết usecase nào đó trong biểu đồ trên.**

***Ví dụ tìm Use Case:***

Nhà băng ABC đưa ra các yêu cầu sau:

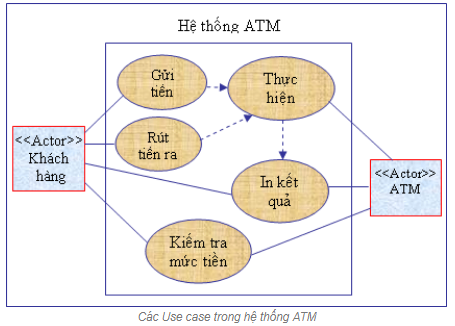
- Một khách hàng có thể muốn gửi tiền vào, rút tiền ra hoặc đơn giản kiểm tra lại số tiền trong tài khoản của anh ta qua máy tự động rút tiền (ATM). Khi đưa tiền vào hoặc rút tiền ra, cần phải ghi ra giấy kết quả những chuyển dịch đã thực hiện và trao tờ giấy này cho khách hàng.

Giải:

- Actor: khách hàng và hệ thống ngân hàng.

- Use Case:

* Gửi tiền vào.
* Rút tiền ra.
* Kiểm tra số tiền trong tài khoản
* Thực hiện các giao dịch nội bộ hệ thống
* In kết quả các giao dịch đã thực hiện.



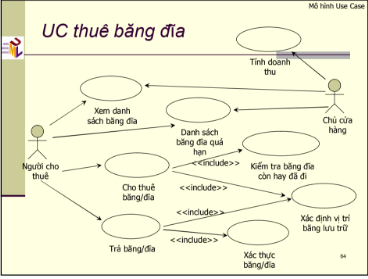
Use Case gửi tiền vào và rút tiền ra phụ thuộc vào Use Case thực hiện các chuyển dịch trong nội bộ hệ thống, việc thực hiện này về phần nó lại phụ thuộc vào chức năng in ra các công việc đã được thực hiện. Kiểm tra mức tiền trong tài khoản là một Use Case độc lập, không phụ thuộc vào các Use Case khác.

*Ví dụ thuê bang đĩa*

* Một của hàng muốn xây dựng hệ thống phần mềm để phục vụ quản lý các hoạt động cho thuê bang đĩa. Mỗi khi khách thuê hỏi mượn thuê hoặc trả bang/đĩa, hệ thống cần phải ghi nhận lại các thông tin và xử lý các công việc thanh toán với khách hàng.
* Ngoài ra hệ thống còn giúp cửa hàng theo dõi doanh thu, kiểm kê các bang đĩa bị hư, còn tốt trong kho, các bang/địa được ưa thích,…

Giải:

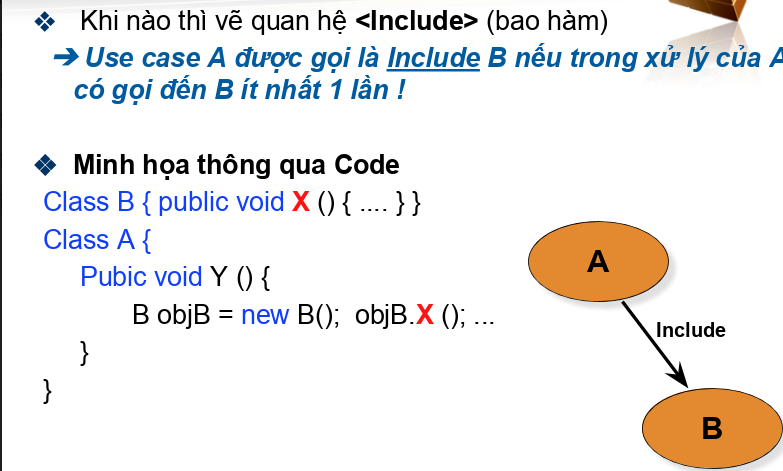
* Danh sách các Actor:
* Khách thuê
* Người cho thuê
* Chủ cửa hàng
* Danh sách các UC:
* Xem danh sách bang đĩa
* Liệt kê danh sách quá hạn
* Cho thuê băng/đĩa
* Trả băng/đĩa
* Sửa danh sách băng đĩa
* Nhập các băng đĩa mới
* Hủy các băng đĩa không còn sử dụng
* Kiểm tra 1 băng hoặc đĩa còn hay đã đi
* Xác định vị trí băng lưu trữ
* Tính doanh thu
* UC thuê băng đĩa:



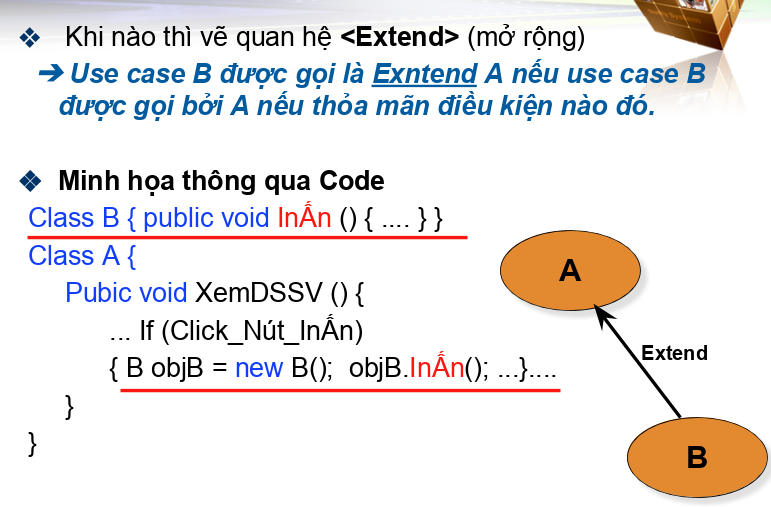
**Đề 5:**

**Câu 1: Anh chị hãy phân biệt các quan hệ include, extend, generation giữa các usecase trong biểu đồ Usecase diagram và cho ví dụ minh họa?**

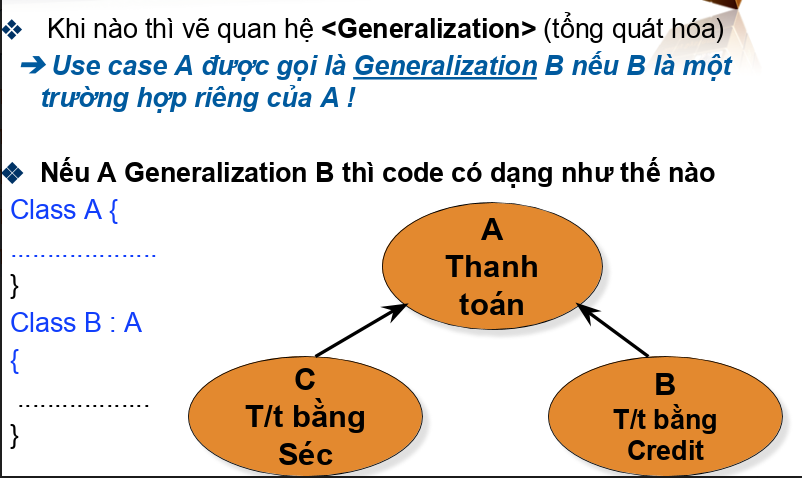
**- Quan hệ <<Include>>:** use case này sử dụng lại chức năng của use case kia.



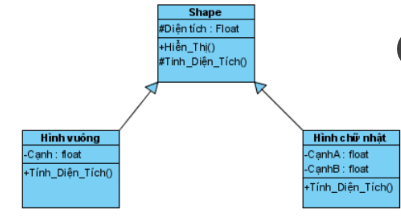
**- Quan hệ <<Extend>>:** use case này mở rộng từ use case kia bằng cách thêm vào một chức năng cụ thể.



**-**



Câu 2: Cho biểu đồ lớp như hình vẽ:



Anh chị hãy phân tích ý nghĩa của biểu đồ trên và minh họa mối quan hệ giữa các class trong mô hình này bằng một ngôn ngữ lập trình hướng đối tượng (JAVA)?

Giải:

* Hình vuông và hình chữ nhật đều Shape để tính diện tích. Do đó, tách hoạt động này thành một lớp và tạo liên kết <<include>> từ hai lớp kia tới nó.
* Mối liên kết (aggregration): Đối tượng Shape được tạo thành từ những đối tượng của Hình vuông và hình chữ nhật.
* Code:

Class Shape{

Protected float Dientich;

Public void Hiển\_Thị();

Protected void Tính\_Diện\_Tích();

}

Class Hìnhvuong include Shape{

Private float Cạnh;

Public void Tính\_Diện\_Tích();

}

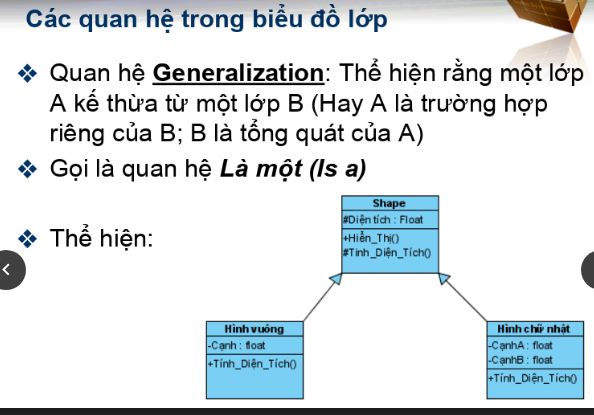
Class Hinhchunhat include Shape{

Private float CanhA;

Private float CanhB;

Public void Tính\_Diện\_Tích();

}



**Câu 1: Trình bày các mối quan hệ giữa các usecase trong biểu đồ usecase? Cho VD và phân tích?**

